

A Symmetry-based Decomposition Approach to Eigenvalue Problems: Formulation, Discretization, and Implementation *

Jun Fang[†] Xingyu Gao[‡] Aihui Zhou[§]

Abstract. In this paper, we propose a decomposition approach for eigenvalue problems with spatial symmetries, including the formulation, discretization as well as implementation. In the formulation, the original problem is divided into a set of subproblems and only a smaller number of eigenpairs is required for each subproblem. This formulation can handle eigenvalue problems with either Abelian or non-Abelian symmetries, and is easy for grid-based discretizations such as finite difference, finite element or finite volume methods. We implement the decomposition approach with finite elements and parallelize our code in two levels. We analyze and illustrate that the decomposition approach can improve the efficiency and scalability of iterative diagonalization. In particular, we apply it to Kohn-Sham equations of symmetric molecules consisting of several hundreds of atoms.

Keywords. Eigenvalue problem, Abelian and non-Abelian symmetry, Group theory, Symmetry-adapted linear combination, Grid-based discretizations, Two-level parallelism.

1 Introduction

Many problems in physics, chemistry and engineering lead to eigenvalue problems of the form

$$Lu = \lambda u \quad \text{in } \Omega \subset \mathbb{R}^n \quad (1.1)$$

subject to some boundary condition, where $n \in \{1, 2, 3\}$ and L is an Hermitian operator. The number of required eigenvalues and eigenfunctions, N_e , could be very large in some applications. For example, in quantum chemistry (see, e.g., [11, 50] and references cited therein), operator L in (1.1) might be the Hamiltonian of a molecular system, and N_e increases with to the number of valence electrons in the system.

Numerical computations of eigenvalues and associated eigenfunctions are usually necessary. It can be very expensive since the computational cost grows in proportion to $N_e^2 N$, where N is size of the discretization. Thus a decomposition of the problem over domain or over

*This work was partially supported by the National Science Foundation of China under Grant 61033009, the Funds for Creative Research Groups of China under Grant 11021101, the National Basic Research Program of China under Grants 2011CB309702 and 2011CB309703, and the National High Technology Research and Development Program of China under Grant 2010AA012303, and the National Center for Mathematics and Interdisciplinary Sciences, CAS.

[†]LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China and Graduate University of Chinese Academy of Sciences, Beijing 100190, China (fangjun@lsec.cc.ac.cn)

[‡]HPCC, Institute of Applied Physics and Computational Mathematics, Beijing 100094, China (gao_xingyu@iapcm.ac.cn)

[§]LSEC, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (azhou@lsec.cc.ac.cn).

eigenvalues is in insistent demand. While it is challenging to decompose eigenvalue problems since they have an intrinsic nonlinearity and can be regarded as global optimization problems with orthonormal constraints. Many kinds of domain decomposition methods derived from boundary value problems do not work effectively for eigenvalue problems.

Symmetry provides a way to do decomposition. Eigenfunctions can be classified into orthogonal categories according to group representation theory. And each category of eigenfunctions can be solved in a subdomain of Ω . Symmetry occurs in everyday life, see, e.g., [8] for a comprehensive illustration of the ubiquitous role of symmetry in nature, music, architecture, and sciences. Mathematically, each symmetry corresponds to an operator, such as a reflection, a rotation or an inversion, that leaves the object or problem invariant. The classification of eigenfunctions comes from their distinct behaviors under the application of symmetry operators.

In quantum physics and quantum chemistry, the so-called symmetry-adapted linear combinations (SALCs) are constructed from specific basis functions like atomic orbitals, internal coordinates of a molecule, or orthogonalized plane waves [8, 14, 15]. Under the SALCs, matrix representation of the problem is block diagonal. Projection operators defined by group theory determine the proper way to combine the original basis functions into SALCs. In [15], the author has given a case-by-case illustration of the way to construct SALCs from atomic orbitals, from which can be seen that the construction of SALCs is not an easy task.

Grid-based discretizations, such as finite difference, finite element and finite volume methods, are widely used in scientific and engineering computations [3, 4, 12, 17, 25]. For instance, the finite element method is important in the discretization of eigenvalue problems in structural analysis [7, 21, 28]. In the last two decades, grid-based discretization approaches have been successfully applied to modern electronic structure calculations, see [6, 16, 40, 45] and reference cited therein. In particular, they have been proven to be well accommodated to petascale computing by treating extremely large-scale eigenvalue problems arising from the electron structure calculations [26, 29].

Grid-based discretizations have good locality and usually result in a large number of degrees of freedom. And finite difference methods do not have basis functions in the classical sense. These facts increase the numerical difficulty to construct SALCs. In this work, we focus on how to use Abelian and non-Abelian symmetries for grid-based discretizations. We propose a decomposition formulation easy for grid-based discretizations, which does not need the explicit construction of SALCs. The original eigenvalue problem is decomposed into a set of subproblems characterized by distinct conditions derived from group representation theory. We also give the discretized subproblems for Abelian and non-Abelian symmetries. For the purpose of comparing our approach with constructing SALCs, we provide mathematical guidance to the construction of SALCs, and then illustrate the equivalence between them by deducing the exact relation of our discretized eigenvalue problem to that formed by SALCs.

We implement the decomposition approach using finite element methods. Subproblems corresponding to different irreducible representations can be solved independently. Accordingly, our code is parallelized in two levels, including a fundamental level of spatial parallelization and another level of subproblem distribution. We apply the approach to solving the Kohn-Sham equation of some cluster systems with symmetries. It is indicated that the decomposition approach would be appreciable for large-scale eigenvalue problems. We should mention that the implementation techniques can be adapted to finite difference and finite volume methods.

We should mention that Bossavit [9, 10] has employed group theory to decompose boundary value problems arising from structural analysis. Note that the design and analysis of decomposition methods for eigenvalue problems is different from boundary value problems, and [5, 10] have pointed out the programming difficulty in the implementation of using symmetries. Kro-

nik and his collaborators [32] have utilized Abelian symmetries in their finite difference codes to simplify the solving of Kohn–Sham equations. However, their implementation is only applicable to Abelian symmetry groups in which any two symmetry operators are commutative. In some plane-wave softwares of electronic structure calculations, symmetries are used to simplify the solving of Kohn–Sham equations by reducing the number of k -points to the irreducible Brillouin zone (IBZ). Two k -points related by some symmetry operation are considered equivalent. For a given k -point, they do not classify the eigenfunctions and thus still solve the original eigenvalue problem.

The rest of this paper is organized as follows. In Section 2, we prove the symmetry-based decomposition of eigenvalue problems, and propose a subproblem formulation proper for grid-based discretizations. Then in Section 3, we give matrix eigenvalue problems derived from the subproblem formulation and deduce the relation to those formed by SALCs, for which purpose we abstract the construction of SALCs mathematically. We quantize the decrease in computational cost when using the decomposition approach in Section 4. And in Section 5 we present some critical implementation issues. In Section 6, we give numerical examples to validate our implementation for Abelian and non-Abelian symmetry groups and show the reduction in computational and communicational overhead; then we apply the decomposition approach to the Kohn–Sham equation in electronic structure calculations. Finally some concluding remarks are given.

2 Decomposition formulation

In this section, we recall some basic but useful results of group theory and reformulate the symmetry-based decomposition for eigenvalue problems to achieve Theorem 2.1. Some notation and concept will be given in Appendix A.

2.1 Representation, basis function, and projection operator

We start the discussion from orthogonal coordinate transformations in \mathbb{R}^n such as a rotation, a reflection or an inversion, that form a finite group G of order g . Denote the function space to be considered as V . Each $R \in G$ corresponds to an operator P_R on function $f \in V$ as

$$P_R f(Rx) = f(x) \quad \forall x \in \mathbb{R}^n.$$

It can be proved that $\{P_R : R \in G\}$ form a group isomorphic to G .

A matrix representation of group G means a group of matrices which is homomorphic to G . Any matrix representation with nonvanishing determinants is equivalent to a representation by unitary matrices (referred to as unitary representation). In the following we focus on unitary representations of group G .

The great orthogonality theorem (cf. [14, 30, 44, 49]) tells that, all the inequivalent, irreducible, unitary representations $\{\Gamma^{(\nu)}\}$ of group G satisfy

$$\sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \Gamma^{(\nu')}(R)_{m'l'} = \delta_{\nu\nu'} \delta_{mm'} \delta_{ll'} \frac{g}{d_\nu}, \quad (2.1)$$

where d_ν denotes the dimensionality of the ν -th representation $\Gamma^{(\nu)}$, $\Gamma^{(\nu)}(R) = (\Gamma^{(\nu)}(R)_{ml})_{d_\nu \times d_\nu}$ and $\Gamma^{(\nu)}(R)_{ml}^*$ is the complex conjugate of $\Gamma^{(\nu)}(R)_{ml}$. The number of all the inequivalent, irreducible, unitary representations is equal to the number of classes in G . We denote this number as n_c .

Definition 2.1. For a given $\Gamma^{(\nu)}$, non-zero functions $\{\phi_1^{(\nu)}, \phi_2^{(\nu)}, \dots, \phi_{d_\nu}^{(\nu)}\} \subset V$ are said to form a basis for $\Gamma^{(\nu)}$ if for any $l \in \{1, 2, \dots, d_\nu\}$

$$P_R \phi_l^{(\nu)} = \sum_{m=1}^{d_\nu} \phi_m^{(\nu)} \Gamma^{(\nu)}(R)_{ml} \quad \forall R \in G. \quad (2.2)$$

$\phi_l^{(\nu)}$ is called to belong to the l -th column of $\Gamma^{(\nu)}$ (or adapt to the ν - l symmetry), and $\{\phi_m^{(\nu)} : m = 1, 2, \dots, d_\nu, m \neq l\}$ are its partners.

There holds an orthogonality property for the basis functions (cf. [44]): if $\{\phi_l^{(\nu)} : l = 1, 2, \dots, d_\nu\}$ and $\{\psi_{l'}^{(\nu')}\} : l' = 1, 2, \dots, d_{\nu'}$ are basis functions for irreducible representations $\Gamma^{(\nu)}$ and $\Gamma^{(\nu')}$, respectively, then the scalar product

$$(\phi_l^{(\nu)}, \psi_{l'}^{(\nu')}) = \delta_{\nu\nu'} \delta_{ll'} d_\nu^{-1} \sum_{m=1}^{d_\nu} (\phi_m^{(\nu)}, \psi_m^{(\nu')}) \quad \forall l \in \{1, 2, \dots, d_\nu\}, l' \in \{1, 2, \dots, d_{\nu'}\}. \quad (2.3)$$

This equation implies that, two functions which belong to different irreducible representations or to different columns of the same unitary representation are orthogonal. What is more, the scalar product of two functions belonging to the same column of a given unitary representation (or adapting to the same symmetry) is independent of the column label.

Multiply equation (2.2) by $\Gamma^{(\nu')}(R)_{m'l'}^*$ and sum over R , the great orthogonality theorem (2.1) implies that

$$\sum_{R \in G} \Gamma^{(\nu')}(R)_{m'l'}^* P_R \phi_l^{(\nu)} = \delta_{\nu\nu'} \delta_{ll'} \frac{g}{d_\nu} \phi_{m'}^{(\nu')}.$$

Define for any $m, l \in \{1, 2, \dots, d_\nu\}$ operator $\mathcal{P}_{ml}^{(\nu)}$ as

$$\mathcal{P}_{ml}^{(\nu)} = \frac{d_\nu}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* P_R, \quad (2.4)$$

then we have

$$\mathcal{P}_{ml}^{(\nu)} \phi_{l'}^{(\nu')} = \delta_{\nu\nu'} \delta_{ll'} \phi_m^{(\nu)}. \quad (2.5)$$

Proposition 2.1. Given $\nu \in \{1, 2, \dots, n_c\}$ and $k \in \{1, 2, \dots, d_\nu\}$. If function $v \in V$ satisfies $\mathcal{P}_{kk}^{(\nu)} v \neq 0$, then $\{\mathcal{P}_{lk}^{(\nu)} v : l = 1, 2, \dots, d_\nu\}$ form a basis for $\Gamma^{(\nu)}$, namely, $\{\mathcal{P}_{lk}^{(\nu)} v : l = 1, 2, \dots, d_\nu\}$ are non-zero functions, and for any $l \in \{1, 2, \dots, d_\nu\}$,

$$P_R \left(\mathcal{P}_{lk}^{(\nu)} v \right) = \sum_{m=1}^{d_\nu} \left(\mathcal{P}_{mk}^{(\nu)} v \right) \Gamma^{(\nu)}(R)_{ml} \quad \forall R \in G. \quad (2.6)$$

Proof. For any $l \in \{1, 2, \dots, d_\nu\}$,

$$P_R \left(\mathcal{P}_{lk}^{(\nu)} v \right) = \frac{d_\nu}{g} \sum_{S \in G} \Gamma^{(\nu)}(S)_{lk}^* P_{RS} v = \frac{d_\nu}{g} \sum_{S' \in G} \Gamma^{(\nu)}(R^{-1} S')_{lk}^* P_{S'} v \quad \forall R \in G,$$

where $P_R P_S = P_{RS}$ because $\{P_R : R \in G\}$ form a group isomorphic to G .

Since $\Gamma^{(\nu)}$ is a unitary representation of G , we obtain

$$\begin{aligned} P_R \left(\mathcal{P}_{lk}^{(\nu)} v \right) &= \frac{d_\nu}{g} \sum_{S' \in G} \left(\sum_{m=1}^{d_\nu} \Gamma^{(\nu)}(R)_{ml} \Gamma^{(\nu)}(S')_{mk}^* \right) P_{S'} v \\ &= \sum_{m=1}^{d_\nu} \left(\mathcal{P}_{mk}^{(\nu)} v \right) \Gamma^{(\nu)}(R)_{ml} \quad \forall R \in G. \end{aligned}$$

Recall the way to achieve (2.5), we see from this equation and the great orthogonality theorem that

$$\mathcal{P}_{kk}^{(\nu)} v = \mathcal{P}_{kl}^{(\nu)} \left(\mathcal{P}_{lk}^{(\nu)} v \right) \quad \forall l \in \{1, 2, \dots, d_\nu\}.$$

So $\mathcal{P}_{kk}^{(\nu)} v \neq 0$ indicates $\mathcal{P}_{lk}^{(\nu)} v \neq 0$ for all $l \in \{1, 2, \dots, d_\nu\}$. This completes the proof. \square

If we set $\nu' = \nu$, $l' = l$ and $m = l$ in (2.5), then it gives

$$\mathcal{P}_{ll}^{(\nu)} \phi_l^{(\nu)} = \phi_l^{(\nu)}. \quad (2.7)$$

Proposition 2.1 implies that (2.7) serves to characterize the labels of any basis function:

Corollary 2.1. *Given $\nu \in \{1, 2, \dots, n_c\}$ and $l \in \{1, 2, \dots, d_\nu\}$. Non-zero function $v \in V$ belongs to the l -th column of $\Gamma^{(\nu)}$ (or adapts to the ν - l symmetry) if and only if*

$$\mathcal{P}_{ll}^{(\nu)} v = v. \quad (2.8)$$

We will use the following properties of operator $\mathcal{P}_{ml}^{(\nu)}$, whose proof is given in Appendix B.

Proposition 2.2. (a) *The adjoint of operator $\mathcal{P}_{ml}^{(\nu)}$*

$$\mathcal{P}_{ml}^{(\nu)*} = \mathcal{P}_{lm}^{(\nu)}. \quad (2.9)$$

(b) *The multiplication of two operators*

$$\mathcal{P}_{ml}^{(\nu)} \mathcal{P}_{m'l'}^{(\nu')} = \delta_{\nu\nu'} \delta_{lm'} \mathcal{P}_{ml}^{(\nu)}. \quad (2.10)$$

2.2 Subproblems

It can be seen from Corollary 2.1 and the linearity of operator $\mathcal{P}_{ll}^{(\nu)}$ that, all functions in V belonging to the l -th column of $\Gamma^{(\nu)}$ (or adapting to the ν - l symmetry) form a subspace of V . We denote this subspace by $V_l^{(\nu)}$.

There holds a decomposition theorem for any function in V (cf. [44]): any function $f \in V$ can be decomposed into a sum of the form

$$f = \sum_{\nu=1}^{n_c} \sum_{l=1}^{d_\nu} f_l^{(\nu)}, \quad (2.11)$$

where $f_l^{(\nu)}$ belongs to subspace $V_l^{(\nu)}$. We see from (2.5) and (2.11) that $\mathcal{P}_{ll}^{(\nu)} : V \rightarrow V_l^{(\nu)}$ is a projection operator. Equation (2.11) implies

$$V = \sum_{\nu=1}^{n_c} \sum_{l=1}^{d_\nu} V_l^{(\nu)},$$

which indeed is a direct sum

$$V = \bigoplus_{\nu=1}^{n_c} \bigoplus_{l=1}^{d_\nu} V_l^{(\nu)} \quad (2.12)$$

due to (2.3).

Now we turn to study the symmetry-based decomposition for eigenvalue problems. Suppose group G is a symmetry group associated with eigenvalue problem (1.1), namely

$$R\Omega = \Omega, \quad P_R L = L P_R \quad \forall R \in G, \quad (2.13)$$

and the subjected boundary condition is also invariant under $\{P_R\}$. Then any $R \in G$ is called a symmetry operation for problem (1.1). For simplicity, we take zero boundary condition as an example and discuss the decomposition of eigenvalue problem

$$\begin{cases} Lu = \lambda u & \text{in } \Omega \subset \mathbb{R}^n, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (2.14)$$

Since P_R and L are commutative for any R in G , we have:

Proposition 2.3. *If function $v \in V_l^{(\nu)}$, then $Lv \in V_l^{(\nu)}$, where $\nu \in \{1, 2, \dots, n_c\}$ and $l \in \{1, 2, \dots, d_\nu\}$. In other words, $V_l^{(\nu)}$ is an invariant subspace of operator L .*

The direct sum decomposition of space V and Proposition 2.3 indicate a decomposition of the eigenvalue problem.

Theorem 2.1. *Suppose system (2.14) has a symmetry group $G = \{R\}$ with a finite order. Denote all the inequivalent, irreducible, unitary representations of G as $\{\Gamma^{(\nu)} : \nu = 1, 2, \dots, n_c\}$. Then the eigenvalue problem can be decomposed into $\sum_{\nu=1}^{n_c} d_\nu$ subproblems. For any $\nu \in \{1, 2, \dots, n_c\}$, the corresponding d_ν subproblems are*

$$\begin{cases} Lu_l^{(\nu)} = \lambda^{(\nu)} u_l^{(\nu)} & \text{in } \Omega, \\ u_l^{(\nu)} = 0 & \text{on } \partial\Omega, \\ u_l^{(\nu)} = \mathcal{P}_{lk}^{(\nu)} u_k^{(\nu)} & \text{in } \Omega, \end{cases} \quad l = 1, 2, \dots, d_\nu, \quad (2.15)$$

where k is any chosen number in $\{1, 2, \dots, d_\nu\}$.

Proof. It can be seen from (2.12) and Proposition 2.3 that, other than solving the eigenvalue problem in V , we can solve the problem in each subspace $V_l^{(\nu)}$ independently. The original eigenvalue problem (2.14) can be decomposed into $\sum_{\nu=1}^{n_c} d_\nu$ subproblems as follows

$$\begin{cases} Lu_l^{(\nu)} = \lambda_l^{(\nu)} u_l^{(\nu)} & \text{in } \Omega, \\ u_l^{(\nu)} = 0 & \text{on } \partial\Omega, \\ \mathcal{P}_{ll}^{(\nu)} u_l^{(\nu)} = u_l^{(\nu)} & \text{in } \Omega, \end{cases} \quad \nu = 1, 2, \dots, n_c, \quad l = 1, 2, \dots, d_\nu, \quad (2.16)$$

where the third equation characterizes $u_l^{(\nu)} \in V_l^{(\nu)}$, as indicated in Corollary 2.1.

Given any $\nu \in \{1, 2, \dots, n_c\}$, we consider the d_ν subproblems in (2.16). We shall prove that, for any chosen $k \in \{1, 2, \dots, d_\nu\}$, if v and w are two orthogonal eigenfunctions corresponding to some eigenvalue of the k -th subproblem, then $\mathcal{P}_{lk}^{(\nu)} v$ and $\mathcal{P}_{lk}^{(\nu)} w$ are eigenfunctions of the l -th subproblem with the same eigenvalue, and are also orthogonal.

From Equation (2.5) and the fact that P_R and L are commutative for each R , we see $\mathcal{P}_{lk}^{(\nu)}v$ is an eigenfunction of the l -th subproblem which corresponds to the same eigenvalue as the one for v . It remains to prove the orthogonality of $\mathcal{P}_{lk}^{(\nu)}v$ and $\mathcal{P}_{lk}^{(\nu)}w$. Proposition 2.2 indicates that the scalar product of any two functions in $V_k^{(\nu)}$ is invariant after operating on them with $\mathcal{P}_{lk}^{(\nu)}$. Indeed, we have for any $v, w \in V_k^{(\nu)}$ that

$$\begin{aligned} (\mathcal{P}_{lk}^{(\nu)}v, \mathcal{P}_{lk}^{(\nu)}w) &= (\mathcal{P}_{lk}^{(\nu)*} \mathcal{P}_{lk}^{(\nu)}v, w) \\ &= (\mathcal{P}_{kl}^{(\nu)} \mathcal{P}_{lk}^{(\nu)}v, w) \\ &= (\mathcal{P}_{kk}^{(\nu)}v, w) \\ &= (v, w), \end{aligned}$$

where the last equality can be obtained from Corollary 2.1. Thus $\mathcal{P}_{lk}^{(\nu)}v$ and $\mathcal{P}_{lk}^{(\nu)}w$ are orthogonal when v and w are.

Recall L is Hermitian, we obtain that eigenvalues of the d_ν subproblems are the same, and eigenfunctions of the l -th subproblem can be chosen as $\{\mathcal{P}_{lk}^{(\nu)}v\}$, where $\{v\}$ are eigenfunctions of the k -th subproblem, and k is any chosen number in $\{1, 2, \dots, d_\nu\}$.

Therefore, the original eigenvalue problem (2.14) can be decomposed into $\sum_{\nu=1}^{n_c} d_\nu$ subproblems, and for any $\nu \in \{1, 2, \dots, n_c\}$ the corresponding d_ν subproblems can be written in the form shown in (2.15). This completes the proof. \square

The third equation of the d_ν subproblems in (2.15) are

$$\begin{aligned} u_k^{(\nu)} &= \mathcal{P}_{kk}^{(\nu)}u_k^{(\nu)}, \\ u_l^{(\nu)} &= \mathcal{P}_{lk}^{(\nu)}u_k^{(\nu)} \quad \forall l = 1, 2, \dots, d_\nu, l \neq k. \end{aligned}$$

We see from Proposition 2.1 that $\{u_l^{(\nu)} : l = 1, 2, \dots, d_\nu\}$ form a basis for $\Gamma^{(\nu)}$. Namely, for any $l \in \{1, 2, \dots, d_\nu\}$

$$P_R u_l^{(\nu)}(x) = \sum_{m=1}^{d_\nu} u_m^{(\nu)}(x) \Gamma^{(\nu)}(R)_{ml} \quad \forall R \in G.$$

Corollary 2.2. *Under the same condition as in Theorem 2.1, eigenvalue problem (2.14) can be decomposed into $\sum_{\nu=1}^{n_c} d_\nu$ subproblems. For any $\nu \in \{1, 2, \dots, n_c\}$, the corresponding d_ν subproblems can be given as follows*

$$\left\{ \begin{array}{ll} Lu_l^{(\nu)} = \lambda^{(\nu)} u_l^{(\nu)} & \text{in } \Omega, \\ u_l^{(\nu)} = 0 & \text{on } \partial\Omega, \\ P_R u_l^{(\nu)}(x) = \sum_{m=1}^{d_\nu} u_m^{(\nu)}(x) \Gamma^{(\nu)}(R)_{ml} & \text{in } \Omega, \forall R, \end{array} \right. \quad l = 1, 2, \dots, d_\nu. \quad (2.17)$$

The third equations in (2.15) and (2.17) play a critical role in the symmetry-based decomposition. The original eigenvalue problem can be decomposed into subproblems just because eigenfunctions of subproblems satisfy distinct equations. In the following text, we call these equations as symmetry characteristics.

Remark 2.1. *Symmetry characteristics describe symmetry properties of eigenfunctions over domain Ω . In [10], the author has given a decomposition formulation for boundary value problems with spatial symmetries. Each decomposed problem is characterized by a “boundary condition” on symmetry element¹ Σ_g , which is in fact a restriction of the symmetry characteristic on Σ_g .*

In our opinion, symmetry characteristics are essential conditions and should not be replaced by the restriction on symmetry elements. In some cases, indeed, boundary conditions such as Dirichlet or Neumann kind on Σ_g can be deduced, while in the derivation one needs to use the symmetry characteristic near Σ_g , not only on it. In some other cases, symmetry characteristics may not derive proper boundary conditions. Therefore, boundary conditions are not sufficient for the well-posedness of subproblems.

Let us compare eigenvalue structures of the original problem and subproblems.

We assume for instance that all symmetries of the original problem form a D_4 group. The D_4 group have $n_c = 5$ irreducible representations; one of them is two-dimensional and all others are one-dimensional [14, 44]. Then the eigenvalues are either nondegenerate or have double degeneracy, on the assumption that no accidental degeneracy occurs [44, 49]. According to Theorem 2.1, the original problem can be decomposed into $\sum_{\nu=1}^{n_c} d_\nu = 6$ subproblems. Any eigenvalue with double degeneracy is distributed to the two subproblems corresponding to ν with $d_\nu = 2$. So all eigenvalues of each subproblem should be nondegenerate.

Actually, the third equations in subproblem formulations (2.15) and (2.17) imply a relation between symmetry and degeneracy. Indeed, for instance, [33, 37, 46] discussed this relation for eigenmodes of the Laplacian in two dimensions. In practice, we usually use some subgroup of the whole symmetry group. Thus subproblems will probably still have degenerate eigenvalues. However, there is a possibility to enlarge the gap of the spectrum, especially when we exploit as many symmetries as possible. This would benefit the convergence of iterative diagonalization.

Formulation (2.17) makes a straightforward implementation for grid-based discretizations. We shall discuss the way to solve the subproblems in the next section.

3 Discretization

In this section, we represent subproblems (2.17) under grid-based discretizations. We present a mathematical guidance to the construction of SALCs in Proposition 3.1 and Remark 3.2. And then we illustrate the relation of our discretized systems to those formed by SALCs.

Note that the d_ν subproblems associated with different ν values are independent and have the same formulations. So we take one $\nu \in \{1, 2, \dots, n_c\}$ and discuss corresponding d_ν subproblems.

3.1 Discretized system

Suppose Ω is discretized by a symmetrical grid associated with group G , and the grid contains N degrees of freedom. For simplicity we assume that no grid points lies on symmetry elements.

We determine a smallest set of degrees of freedom that could produce all N ones by applying symmetry operators $\{R \in G\}$. It is clear that number of degrees of freedom in this smallest set

¹ Symmetry element of operation R is a point of reference about which R is carried out, such as a point to do inversion, a rotation axis, or a reflection plane. It is seen that symmetry element is invariant under the associated symmetry operation.

$N_0 = \frac{1}{g}N$. We denote them as

$$\{x_j : j = 1, 2, \dots, N_0\},$$

then all degrees of freedom can be given by

$$\{R(j) : j = 1, 2, \dots, N_0, R \in G\},$$

where $R(j) \equiv Rx_j$ ($j = 1, 2, \dots, N_0$).

The symmetry characteristic equation in (2.17) tells that for any $l \in \{1, 2, \dots, d_\nu\}$, $u_l^{(\nu)}$ over Ω is determined by the values of functions $\{u_1^{(\nu)}, \dots, u_{d_\nu}^{(\nu)}\}$ over Ω_0 . Under discretization it is, for any $l \in \{1, 2, \dots, d_\nu\}$, the values of $u_l^{(\nu)}$ on all degrees of freedom $\{R(j) : j = 1, 2, \dots, N_0, R \in G\}$ are determined by the values of functions $\{u_1^{(\nu)}, \dots, u_{d_\nu}^{(\nu)}\}$ on $\{j : j = 1, 2, \dots, N_0\}$. Thus, the size of discretized eigenvalue problem for (2.17) is $d_\nu N_0$.

If the given irreducible representation $\Gamma^{(\nu)}$ is one-dimensional, then (2.17) gives

$$\begin{cases} Lu^{(\nu)} = \lambda^{(\nu)} u^{(\nu)} & \text{in } \Omega, \\ u^{(\nu)} = 0 & \text{on } \partial\Omega, \\ u^{(\nu)}(Rx) = \Gamma^{(\nu)}(R)^* u^{(\nu)}(x) & \text{in } \Omega, \forall R, \end{cases} \quad (3.1)$$

where we omit subscripts of $\Gamma^{(\nu)}(R)_{11}$ and $u_1^{(\nu)}$.

Suppose the discretized system for eigenvalue problem (3.1) are

$$\sum_{j=1}^{N_0} \sum_{R \in G} a_{i,R(j)} u_{R(j)} = \lambda^{(\nu)} \sum_{j=1}^{N_0} \sum_{R \in G} b_{i,R(j)} u_{R(j)} \quad \forall i = 1, 2, \dots, N_0.$$

We know from the symmetry characteristic that the discretized system can be reduced to

$$\sum_{j=1}^{N_0} \sum_{R \in G} \Gamma^{(\nu)}(R)^* a_{i,R(j)} u_j = \lambda^{(\nu)} \sum_{j=1}^{N_0} \sum_{R \in G} \Gamma^{(\nu)}(R)^* b_{i,R(j)} u_j \quad \forall i = 1, 2, \dots, N_0.$$

Denote the solution vector as

$$\mathbf{u} = (u_1, u_2, \dots, u_{N_0})^\top.$$

We write the discretized system into the matrix form

$$A\mathbf{u} = \lambda^{(\nu)} B\mathbf{u},$$

where

$$\begin{aligned} A &= (A_{ij})_{N_0 \times N_0}, & A_{ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)^* a_{i,R(j)}, \\ B &= (B_{ij})_{N_0 \times N_0}, & B_{ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)^* b_{i,R(j)}. \end{aligned} \quad (3.2)$$

In the case of higher-dimensional irreducible representations, the d_ν subproblems in (2.17) are coupled through symmetry characteristics.

Taking $d_\nu = 2$ as an example, we assemble subproblems for $u_1^{(\nu)}$ and $u_2^{(\nu)}$ in (2.17) to solve eigenvalue problem

$$\left\{ \begin{array}{ll} \begin{pmatrix} Lu_1^{(\nu)} \\ Lu_2^{(\nu)} \end{pmatrix} = \lambda^{(\nu)} \begin{pmatrix} u_1^{(\nu)} \\ u_2^{(\nu)} \end{pmatrix} & \text{in } \Omega, \\ \begin{pmatrix} u_1^{(\nu)} \\ u_2^{(\nu)} \end{pmatrix} (Rx) = \begin{pmatrix} \Gamma^{(\nu)}(R)_{11}^* & \Gamma^{(\nu)}(R)_{12}^* \\ \Gamma^{(\nu)}(R)_{21}^* & \Gamma^{(\nu)}(R)_{22}^* \end{pmatrix} \begin{pmatrix} u_1^{(\nu)} \\ u_2^{(\nu)} \end{pmatrix} (x) & \text{in } \Omega, \forall R, \\ \begin{pmatrix} u_1^{(\nu)} \\ u_2^{(\nu)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{on } \partial\Omega. \end{array} \right. \quad (3.3)$$

Suppose the discretized system for (3.3) are

$$\left\{ \begin{array}{ll} \sum_{j=1}^{N_0} \sum_{R \in G} a_{i,R(j)} u_{1,R(j)} = \lambda^{(\nu)} \sum_{j=1}^{N_0} \sum_{R \in G} b_{i,R(j)} u_{1,R(j)} & \forall i = 1, 2, \dots, N_0, \\ \sum_{j=1}^{N_0} \sum_{R \in G} a_{i,R(j)} u_{2,R(j)} = \lambda^{(\nu)} \sum_{j=1}^{N_0} \sum_{R \in G} b_{i,R(j)} u_{2,R(j)} & \forall i = 1, 2, \dots, N_0. \end{array} \right. \quad (3.4)$$

Denote the solution vector as

$$\mathbf{v} = (u_{11}, u_{12}, \dots, u_{1N_0}, u_{21}, u_{22}, \dots, u_{2N_0})^\top$$

and write the discretized system in a matrix form

$$A\mathbf{v} = \lambda^{(\nu)} B\mathbf{v}.$$

Then

$$A = \begin{pmatrix} A_{[11]} & A_{[12]} \\ A_{[21]} & A_{[22]} \end{pmatrix}, \quad B = \begin{pmatrix} B_{[11]} & B_{[12]} \\ B_{[21]} & B_{[22]} \end{pmatrix},$$

where $A_{[ml]} = (A_{[ml]ij})_{N_0 \times N_0}$ ($m, l = 1, 2$) with

$$\begin{aligned} A_{[11]ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)_{11}^* a_{i,R(j)}, & A_{[12]ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)_{12}^* a_{i,R(j)}, \\ A_{[21]ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)_{21}^* a_{i,R(j)}, & A_{[22]ij} &= \sum_{R \in G} \Gamma^{(\nu)}(R)_{22}^* a_{i,R(j)}. \end{aligned} \quad (3.5)$$

Matrix elements of B are in the same form as those of A and can be obtained by substituting $a_{i,R(j)}$ with $b_{i,R(j)}$.

Remark 3.1. *If symmetry group G is Abelian, each irreducible representation is one-dimensional and all discretized subproblems are independent. Otherwise, there exist $\Gamma^{(\nu)}$ with $d_\nu > 1$ and the corresponding d_ν discretized subproblems are coupled through symmetry characteristics. Thus, no matter G is Abelian or not, we shall solve n_c decoupled eigenvalue subproblems, where n_c is the number of irreducible representations. And the size of discretized system for the ν -th problem is $d_\nu N_0$.*

3.2 Finite element discretization

In this part, taking the finite element discretization as an example, we study the relation between our discretized systems and those from constructing SALCs. We propose Proposition 3.1 and Remark 3.2 to guide the construction of SALCs from mathematical viewpoint.

Suppose the weak form of (2.14) is: Find $(\lambda, u) \in \mathbb{R} \times V$ such that

$$a(u, v) = \lambda(u, v) \quad \forall v \in V. \quad (3.6)$$

Consider the finite element discretization, we denote the basis function corresponding to any $j \in \{1, 2, \dots, N_0\}$ as φ_j . It can be seen from $P_R \varphi_j(x) = \varphi_j(R^{-1}x)$ that $P_R \varphi_j$ is the basis function corresponding to $R(j)$, i.e.,

$$P_R \varphi_j = \varphi_{R(j)}.$$

In this case, our discretized systems are determined by setting $a_{i,R(j)}$ and $b_{i,R(j)}$ in (3.2) and (3.5) as

$$a_{i,R(j)} = a(P_R \varphi_j, \varphi_i), \quad b_{i,R(j)} = (P_R \varphi_j, \varphi_i). \quad (3.7)$$

We shall study the relation between our discretized systems with those from SALCs. First we discuss more about the construction of SALCs.

For the given ν , we fix some $k \in \{1, 2, \dots, d_\nu\}$ and generate SALCs for the k -th subproblem in (2.15). This is achieved by applying projection operator $\mathcal{P}_{kk}^{(\nu)}$ on all the finite element basis functions $\{P_R \varphi_j : j = 1, 2, \dots, N_0, R \in G\}$. Suppose we obtain N' linearly independent basis functions from this process and we denote them as $\{\Psi_j : j = 1, 2, \dots, N'\}$. Then for any $l \in \{1, 2, \dots, d_\nu\}$, SALCs for the l -th subproblem can be given as $\{\mathcal{P}_{lk}^{(\nu)} \Psi_j : j = 1, 2, \dots, N'\}$.

Consider the d_ν discretized systems under the generated SALCs. Matrix elements of the l -th discretized system are

$$a(\mathcal{P}_{lk}^{(\nu)} \Psi_j, \mathcal{P}_{lk}^{(\nu)} \Psi_i), \quad (\mathcal{P}_{lk}^{(\nu)} \Psi_j, \mathcal{P}_{lk}^{(\nu)} \Psi_i) : \quad i, j = 1, 2, \dots, N'.$$

For each $j \in \{1, 2, \dots, N'\}$, functions $\{\mathcal{P}_{lk}^{(\nu)} \Psi_j : l = 1, 2, \dots, d_\nu\}$ form a basis for $\Gamma^{(\nu)}$. We see from (2.3) and Proposition 2.3 that all the d_ν discretized systems are the same. So we only need to solve the discretized system corresponding to the k -th subproblem:

$$\sum_{j=1}^{N'} a(\Psi_j, \Psi_i) c_j = \lambda^{(\nu)} \sum_{j=1}^{N'} (\Psi_j, \Psi_i) c_j \quad i = 1, 2, \dots, N'. \quad (3.8)$$

After calculating $\{c_j\}$, the approximated eigenfunctions can be achieved by

$$u_l^{(\nu)} = \sum_{j=1}^{N'} c_j \mathcal{P}_{lk}^{(\nu)} \Psi_j \quad l = 1, 2, \dots, d_\nu.$$

Now we give a result on how to determine the value of N' , and explain the specific way to obtain these functions.

Proposition 3.1. *Given $\nu \in \{1, 2, \dots, n_c\}$ and $k \in \{1, 2, \dots, d_\nu\}$. If $j \in \{1, 2, \dots, N_0\}$ satisfies that $\{P_R \varphi_j : R \in G\}$ are linearly independent, then there are exactly d_ν linearly independent functions in $\{\mathcal{P}_{kk}^{(\nu)} P_R \varphi_j : R \in G\}$.*

Proof. For any $R \in G$, since

$$\begin{aligned} \mathcal{P}_{kk}^{(\nu)} P_R \varphi_j &= \frac{d_\nu}{g} \sum_{R' \in G} \Gamma^{(\nu)}(R')_{kk}^* P_{R'} P_R \varphi_j \\ &= \frac{d_\nu}{g} \sum_{S \in G} \Gamma^{(\nu)}(SR^{-1})_{kk}^* P_S \varphi_j, \end{aligned} \quad (3.9)$$

we see that $\mathcal{P}_{kk}^{(\nu)} P_R \varphi_j$ is a linear combination of functions $\{P_S \varphi_j : S \in G\}$, and the coefficient of $P_S \varphi_j$ is $\frac{d_\nu}{g} \Gamma^{(\nu)}(SR^{-1})_{kk}^*$.

For the given j , $\{P_S \varphi_j : S \in G\}$ are linearly independent. And R^{-1} runs over all elements of group G when R does. So the number of linearly independent functions in $\{\mathcal{P}_{kk}^{(\nu)} P_R \varphi_j : R \in G\}$ equals to the rank of a matrix $C = (C_{mn})_{g \times g}$, where

$$C_{mn} = \Gamma^{(\nu)}(R_m R_n)^*_{kk}.$$

It can be seen that C_{mn} is the multiplication of the k -th row of $\Gamma^{(\nu)}(R_m)^*$ and the k -th column of $\Gamma^{(\nu)}(R_n)^*$. Then C can be written as

$$C = \begin{pmatrix} \Gamma^{(\nu)}(R_1)_{k1}^* & \cdots & \Gamma^{(\nu)}(R_1)_{kd_\nu}^* \\ \Gamma^{(\nu)}(R_2)_{k1}^* & \cdots & \Gamma^{(\nu)}(R_2)_{kd_\nu}^* \\ \vdots & & \vdots \\ \Gamma^{(\nu)}(R_g)_{k1}^* & \cdots & \Gamma^{(\nu)}(R_g)_{kd_\nu}^* \end{pmatrix} \begin{pmatrix} \Gamma^{(\nu)}(R_1)_{1k}^* & \cdots & \Gamma^{(\nu)}(R_g)_{1k}^* \\ \Gamma^{(\nu)}(R_1)_{2k}^* & \cdots & \Gamma^{(\nu)}(R_g)_{2k}^* \\ \vdots & & \vdots \\ \Gamma^{(\nu)}(R_1)_{d_\nu k}^* & \cdots & \Gamma^{(\nu)}(R_g)_{d_\nu k}^* \end{pmatrix} \equiv C_1 C_2.$$

C_1 and C_2 are $g \times d_\nu$ and $d_\nu \times g$ matrices, respectively. We obtain from the great orthogonality theorem (2.1) that columns of C_1 are orthogonal, and so are rows of C_2 , i.e.,

$$\text{rank}(C_1) = \text{rank}(C_2) = d_\nu.$$

So

$$\text{rank}(C) = \text{rank}(C_1 C_2) = d_\nu,$$

and we have proved the proposition. \square

Remark 3.2. For the given ν and k , if j satisfies some condition, Proposition 3.1 indicates that there are d_ν linearly independent basis functions adapted to the ν - k symmetry in $\{\mathcal{P}_{kk}^{(\nu)} P_R \varphi_j : R \in G\}$. It remains a problem how to obtain the d_ν functions. We see from (3.9) that, whenever the chosen d_ν operators $\{R_n \in G : n = 1, \dots, d_\nu\}$ satisfy that the k -th columns of matrices $\{\Gamma^{(\nu)}(R_n^{-1})^* : n = 1, \dots, d_\nu\}$ are linearly independent, $\{\mathcal{P}_{kk}^{(\nu)} P_{R_n} \varphi_j : n = 1, \dots, d_\nu\}$ exactly give the d_ν linearly independent functions.

In the case of $d_\nu = 1$, we apply projection operator $\mathcal{P}^{(\nu)}$ on all the finite element basis functions to construct the SALCs. We see from Proposition 3.1 that for each $j \in \{1, 2, \dots, N_0\}$, $\{\mathcal{P}^{(\nu)} P_R \varphi_j : R \in G\}$ give one symmetry-adapted basis function. According to Remark 3.2, we can choose $R = E$ to get all the N_0 SALCs as

$$\Phi_j = \mathcal{P}^{(\nu)} \varphi_j, \quad j = 1, 2, \dots, N_0.$$

The discretized system under these SALCs are

$$\sum_{j=1}^{N_0} a(\Phi_j, \Phi_i) c_j = \tilde{\lambda} \sum_{j=1}^{N_0} (\Phi_j, \Phi_i) c_j \quad \forall i = 1, 2, \dots, N_0. \quad (3.10)$$

Equivalently,

$$\tilde{A} \tilde{\mathbf{u}} = \tilde{\lambda} \tilde{B} \tilde{\mathbf{u}}, \quad (3.11)$$

where

$$\tilde{\mathbf{u}} = (c_1, c_2, \dots, c_{N_0})^\top,$$

$$\begin{aligned}\tilde{A} &= (\tilde{A}_{ij})_{N_0 \times N_0}, & \tilde{A}_{ij} &= \frac{1}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)^* a(P_R \varphi_j, \varphi_i), \\ \tilde{B} &= (\tilde{B}_{ij})_{N_0 \times N_0}, & \tilde{B}_{ij} &= \frac{1}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)^* (P_R \varphi_j, \varphi_i).\end{aligned}$$

It can be seen that

$$\tilde{A} = \frac{1}{g} A, \quad \tilde{B} = \frac{1}{g} B.$$

Thus we get

$$\lambda = \tilde{\lambda}, \quad \mathbf{u} = \tilde{\mathbf{u}}. \quad (3.12)$$

In the case of $d_\nu = 2$, there are two subproblems in (2.15). We choose $k = 1$ and apply projection operator $\mathcal{P}_{11}^{(\nu)}$ on all the finite element basis functions to construct SALCs for the first subproblem. Proposition 3.1 tells that for each $j \in \{1, 2, \dots, N_0\}$, $\{\mathcal{P}_{11}^{(\nu)} P_R \varphi_j : R \in G\}$ give $d_\nu = 2$ linearly independent symmetry-adapted basis functions. Based on Remark 3.2, we choose identity operation E and another $S \in G$ which satisfy that the first columns of matrices $\{\Gamma^{(\nu)}(E)^*, \Gamma^{(\nu)}(S^{-1})^*\}$ are linearly independent. Then

$$\{\mathcal{P}_{11}^{(\nu)} \varphi_j, \mathcal{P}_{11}^{(\nu)} P_S \varphi_j : j = 1, 2, \dots, N_0\}$$

give all the $2N_0$ basis functions adapted to the ν -1 symmetry. We denote them as

$$(\Phi_1, \dots, \Phi_{N_0}, \Psi_1, \dots, \Psi_{N_0}) = (\mathcal{P}_{11}^{(\nu)} \varphi_1, \dots, \mathcal{P}_{11}^{(\nu)} \varphi_{N_0}, \mathcal{P}_{11}^{(\nu)} P_S \varphi_1, \dots, \mathcal{P}_{11}^{(\nu)} P_S \varphi_{N_0}).$$

The discretized system under these SALCs are

$$\begin{cases} \sum_{j=1}^{N_0} \{a(\Phi_j, \Phi_i) a_j + a(\Psi_j, \Phi_i) b_j\} = \lambda \sum_{j=1}^{N_0} \{(\Phi_j, \Phi_i) a_j + (\Psi_j, \Phi_i) b_j\} & \forall i = 1, 2, \dots, N_0, \\ \sum_{j=1}^{N_0} \{a(\Phi_j, \Psi_i) a_j + a(\Psi_j, \Psi_i) b_j\} = \lambda \sum_{j=1}^{N_0} \{(\Phi_j, \Psi_i) a_j + (\Psi_j, \Psi_i) b_j\} & \forall i = 1, 2, \dots, N_0, \end{cases} \quad (3.13)$$

Equivalently,

$$\tilde{A} \tilde{\mathbf{v}} = \tilde{\lambda} \tilde{B} \tilde{\mathbf{v}},$$

where

$$\begin{aligned}\tilde{\mathbf{v}} &= (a_1, a_2, \dots, a_{N_0}, b_1, b_2, \dots, b_{N_0})^\top \\ \tilde{A} &= \begin{pmatrix} \tilde{A}_{[11]} & \tilde{A}_{[12]} \\ \tilde{A}_{[21]} & \tilde{A}_{[22]} \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \tilde{B}_{[11]} & \tilde{B}_{[12]} \\ \tilde{B}_{[21]} & \tilde{B}_{[22]} \end{pmatrix}.\end{aligned}$$

It can be calculated that

$$\begin{aligned}\tilde{A}_{[11]} &= \frac{2}{g} A_{[11]}, \\ \tilde{A}_{[12]} &= \frac{2}{g} \left(\Gamma^{(\nu)}(S)_{11} A_{[11]} + \Gamma^{(\nu)}(S)_{12} A_{[12]} \right), \\ \tilde{A}_{[21]} &= \frac{2}{g} \left(\Gamma^{(\nu)}(S)_{11}^* A_{[11]} + \Gamma^{(\nu)}(S)_{12}^* A_{[21]} \right), \\ \tilde{A}_{[22]} &= \frac{2}{g} \left\{ \Gamma^{(\nu)}(S)_{11} \left(\Gamma^{(\nu)}(S)_{11}^* A_{[11]} + \Gamma^{(\nu)}(S)_{12}^* A_{[21]} \right) \right. \\ &\quad \left. + \Gamma^{(\nu)}(S)_{12} \left(\Gamma^{(\nu)}(S)_{11}^* A_{[12]} + \Gamma^{(\nu)}(S)_{12}^* A_{[22]} \right) \right\}.\end{aligned}$$

Let

$$Q_l = \begin{pmatrix} I_{N_0 \times N_0} & \mathbf{0}_{N_0 \times N_0} \\ \Gamma^{(\nu)}(S)_{11}^* I_{N_0 \times N_0} & \Gamma^{(\nu)}(S)_{12}^* I_{N_0 \times N_0} \end{pmatrix}, \quad Q_r = \begin{pmatrix} I_{N_0 \times N_0} & \Gamma^{(\nu)}(S)_{11} I_{N_0 \times N_0} \\ \mathbf{0}_{N_0 \times N_0} & \Gamma^{(\nu)}(S)_{12} I_{N_0 \times N_0} \end{pmatrix},$$

we have

$$Q_l A Q_r = \frac{g}{2} \tilde{A}.$$

Similarly

$$Q_l B Q_r = \frac{g}{2} \tilde{B}.$$

Thus we obtain

$$\lambda = \tilde{\lambda}, \quad \mathbf{v} = Q_r \tilde{\mathbf{v}}, \quad (3.14)$$

i.e.,

$$u_{1j} = a_j + \Gamma^{(\nu)}(S)_{11} b_j, \quad u_{2j} = \Gamma^{(\nu)}(S)_{12} b_j, \quad j = 1, 2, \dots, N_0.$$

Consider a given $\nu \in \{1, 2, \dots, n_c\}$, the method of constructing SALCs seems to have an obvious advantage that the d_ν subproblems are decoupled. Proposition 3.1 tells that the number of SALCs for each subproblem is in fact $d_\nu N_0$. Therefore, the coupled eigenvalue problem appeared in our decomposition approach is not an induced complexity, but some reflection of the intrinsic property of symmetry-based decomposition.

Solving subproblems instead of the original eigenvalue problem shall reduce the computational overhead and memory requirement to a large extent. The eigenvalues to be computed are distributed among subproblems, and the decomposed problems can be solved in a small subdomain. Moreover, as indicated in Section 2, there is a possibility to improve the spectral separation, which would accelerate convergence of iterative diagonalization. Later, we shall address some issues gained from a parallel implementation of the decomposition approach and propose a way to analyze the practical decrease in the computational cost.

4 Complexity and performance analysis

The distinguished advantage of solving subproblems (2.15) instead of the original problem (2.14) is the reduction in computational overhead. Based on a complexity analysis, we quantize this reduction and present a way to analyze the practical speedup in CPU time.

4.1 Complexity analysis

Computational complexity is the dominant part of computational overhead when the size of problem becomes sufficiently large. So the fundamental step of complexity analysis is to figure out the computational cost in floating point operations (flops).

The algebraic eigenvalue problem will be solved by the implicitly restarted Lanczos method (IRLM) implemented in ARPACK package [34]. IRLM is the widely-used and practical variant of Krylov subspace iterative method for computing eigenvalues of large-scale Hermitian matrices. Our complexity analysis will be based on IRLM, whereas it can be extended to other iterative diagonalization methods.

Total flops of the iterative method is the product of the number of iteration steps and number of flops per iteration. Since the number of iterations is problem dependent, we analyze the number of flops per iteration, for which purpose we represent the procedure of IRLM as Algorithm 4.1.

Table 1 is a supplementary remark to Algorithm 4.1. In Algorithm 4.1, Step 2 is the Schur decomposition of H_m , and consumes about $6m^2$ flops [24]. Steps 4 to 7 do l -step QR iteration with shifts. On the assumption that each Q_j is the product of $(m-1)$ Givens transformations, Step 5 costs $8m(m-1)$ flops since applying one Givens transformation to one matrix only

Algorithm 4.1: An implicitly restarted Lanczos method

Input: Maximum number of iteration steps; The m -step Lanczos Factorization

$$AV_m = V_m H_m + f_m e_m^T.$$

```
1 repeat
2   Compute the Schur decomposition of the symmetric tridiagonal matrix  $H_m$  and
   select the set of  $l$  shifts  $\mu_1, \mu_2, \dots, \mu_l$ ;
3    $q^T \leftarrow e_m^T$ ;
4   for  $j = 1, 2, \dots, l$  do
5      $H_m - \mu_j I = Q_j R_j$ ;  $H_m \leftarrow R_j Q_j + \mu_j I$ ;
6      $V_m \leftarrow V_m Q_j$ ;  $q^H \leftarrow q^H Q_j$ ;
7   end
8    $f_k \leftarrow v_{k+1} \hat{\beta}_k + f_m \sigma_k$ ;  $V_k \leftarrow V_m(1 : n, 1 : k)$ ;  $H_k \leftarrow H_m(1 : k, 1 : k)$ ;
9   Beginning with the  $k$ -step Lanczos factorization  $AV_k = V_k H_k + e_k^T f_k$ , apply  $l$ 
   additional steps of the Lanczos process to obtain a new  $m$ -step Lanczos factorization
    $AV_m = V_m H_m + f_m e_m^T$ ;
10 until Convergence or the number of iteration steps exceeded the maximum one;
```

Table 1: Notations in Algorithm 4.1

notation	description
m	the maximum dimension of the Krylov subspace, twice as many as the number of required eigenvalues in our computations
l	the number of Lanczos factorization steps, s.t. $m = k + l$
N	the number of DOFs
A	the (sparse) matrix size of N , arising from the grid-based discretization of (2.14) or (2.15)
V_m	the matrix size of $N \times m$, made of m column vectors as the basis of the Krylov subspace
H_m	the symmetric tridiagonal matrix size of m
f_m	the column vector size of N , the residual vector after m steps of Lanczos factorization
e_m	the m -length unit column vector which m -th component is one
R_j	the upper triangular matrix size of m
Q_j	the unitary matrix size of m
v_{k+1}	the $(k + 1)$ -th column vector of V_m
$\hat{\beta}_k$	$H_m(k + 1, k)$
σ_k	the k -th component of q

change two rows or columns. And for the same reason, Step 6 costs $4(m-1)(N+1)$ flops. Consequently Steps 4 to 7 consumes $4l(m-1)(2m+N+1)$ flops. Regardless of BLAS-1 operations, we do l matrix-vector multiplication operations at Step 9.

Besides matrix size N , the flops of one matrix-vector multiplication also depend on the order of finite difference or finite elements. If the shift-invert mode in ARPACK is applied to solve the general eigenvalue problems arising from the finite element discretization, the matrix-vector multiplication will be realized by some iterative linear solver. So we cannot figure out accurately the flops per matrix-vector multiplication but represent it as $\mathcal{O}(N)$.

In total, the computational overhead per IRLM iteration can be estimated as

$$6m^2 + l[4(m-1)(2m+N+1) + \mathcal{O}(N)]$$

flops. In general, the matrix size N is much more than m in either finite difference or finite element discretizations. So the majority of the flops per IRLM iteration is

$$f(l, m, N) = l[4mN + \mathcal{O}(N)]. \quad (4.1)$$

In order to make clear the reduction in flops per iteration, we divide flops per iteration into two parts. One is required by l -step QR iteration, and the other is spent on l operations of matrix-vector multiplication. We denote them by f_1 and f_2 respectively and rewrite (4.1) as follows

$$\begin{aligned} f(l, m, N) &= f_1(l, m, N) + f_2(l, m, N), \\ f_1(l, m, N) &\equiv 4lmN, \\ f_2(l, m, N) &\equiv \mathcal{O}(lN). \end{aligned} \quad (4.2)$$

In solving the original eigenvalue problem (2.14), the major flops per IRLM iteration can be accounted as (4.1) or (4.3). In solving one subproblem (2.15), m is reduced to m/d_1 , N is reduced to N/g , l is replaced by l/θ_2 , where g is given in (A.1), $\theta_1 > 1$ and $\theta_2 \approx \theta_1$ as l is almost proportional to m in Algorithm 4.1. Due to $\sum_{\nu=1}^{n_c} d_\nu$ subproblems in total and the identical θ_1 for each subproblem (2.15), we see that the majority of total flops per iteration for all the subproblem is

$$\begin{aligned} n_p f\left(\frac{l}{\theta_2}, \frac{m}{\theta_1}, \frac{N}{g}\right) &= n_p \left[f_1\left(\frac{l}{\theta_2}, \frac{m}{\theta_1}, \frac{N}{g}\right) + f_2\left(\frac{l}{\theta_2}, \frac{m}{\theta_1}, \frac{N}{g}\right) \right] \\ &= \frac{n_p}{g} \left[\frac{1}{\theta_1 \theta_2} f_1(l, m, N) + \frac{1}{\theta_2} f_2(l, m, N) \right], \end{aligned} \quad (4.3)$$

where the number of subproblems

$$n_p = \sum_{\nu=1}^{n_c} d_\nu. \quad (4.4)$$

As mentioned in Section 3, it can be qualitatively said that the decomposition approach saves the computational cost of solving the eigenvalue problem. Now the reduction can be quantized by (4.3).

4.2 Performance analysis

Since the order of factors for f_1 and f_2 is not identical in (4.3), the practical speedup in CPU time cannot be properly quantized by (4.3). Thus, in solving the original eigenvalue problem

(2.14), we introduce the CPU time ratio ω of the matrix-vector multiplications to the whole IRLM process. It is an a posteriori parameter which screens affects of implementation, runtime environment, as well as the specific linear solver for the shift-invert mode. Besides, testing for ω is feasible as the operation of the matrix-vector multiplication is usually provided by users.

Applying the symmetry-based decomposition approach instead of solving (2.14) directly, we can explain the speedup in CPU time of one iteration as follows:

$$\begin{aligned} s(\theta_1, \theta_2, \omega) &= \frac{\frac{1}{\omega}}{\frac{n_p}{g} \left(\frac{1}{\theta_1 \theta_2} \frac{1-\omega}{\omega} + \frac{1}{\theta_2} \right)} \\ &= \frac{g\theta_1\theta_2}{n_p [1 + (\theta_1 - 1)\omega]} \end{aligned} \quad (4.5)$$

$$\approx \frac{g\theta_1^2}{n_p [1 + (\theta_1 - 1)\omega]}. \quad (4.6)$$

In practice, θ_2 is actually determined by the internal configurations of algebraic eigenvalue solvers. So we prefer (4.6) and use it to predict the CPU time speedup before solving subproblems (2.15).

In Section 6, the validation of (4.5) will be well supported by numerical experiments. Moreover, this performance analysis implies that the speedup will be amplified while more required eigenvalues are wanted and a consequent grow in ω is very likely. Therefore, the symmetry-based decomposition will be attractive for large-scale eigenvalue problems.

5 Practical issues

In this section, we address some key issues on implementation.

5.1 Implementation of symmetry characteristics

Symmetry characteristics play a critical role in the decomposition approach, so it is important to preserve and realize symmetry characteristics for discretized eigenfunctions.

For all the degrees of freedom not lying on symmetry elements, the implementation of symmetry characteristics is straightforward with grid discretization. If $x \in \Omega$ is a grid point lying on the symmetry element corresponding to operation $R \in G$, the symmetry characteristic

$$u_l^{(\nu)}(Rx) = \sum_{m=1}^{d_\nu} \Gamma^{(\nu)}(R)_{lm}^* u_m^{(\nu)}(x)$$

reduces to

$$u_l^{(\nu)}(x) = \sum_{m=1}^{d_\nu} \Gamma^{(\nu)}(R)_{lm}^* u_m^{(\nu)}(x).$$

If it holds that

$$\det \left(\Gamma^{(\nu)}(R) - I_{d_\nu \times d_\nu} \right) \neq 0,$$

all values $u_1^{(\nu)}(x), \dots, u_{d_\nu}^{(\nu)}(x)$ are zeros. Otherwise, we have to find the independent ones out of $u_1^{(\nu)}(x), \dots, u_{d_\nu}^{(\nu)}(x)$ and treat them as the other degrees of freedom.

In our computation, we discretize the problem on a tensor-product grid associated with the symmetry group. Currently, for simplicity, we just avoid symmetry elements on coordinate

planes, by imposing an odd number of partition in each direction and using finite elements of odd orders.

5.2 Distribution of required eigenvalues among subproblems

The required eigenvalues of the original eigenvalue problem (2.14) are distributed among associated subproblems, and the number of eigenvalues required by each subproblem can be almost reduced by as many times as the number of subproblems. However, we cannot know in advance the symmetry properties of eigenfunctions corresponding to required eigenvalues. Thus we have to consider some redundant eigenvalues for each subproblems.

We suppose to solve the first m eigenvalues of the original eigenvalue problem. First we set the number of eigenvalues to be computed for each subproblem as $\frac{m}{\sum_{\nu=1}^{n_c} d_\nu}$ plus redundant m_a eigenvalues. After solving the subproblems, we gather the eigenvalues from all subproblems and sort them in the ascending order. After taking m smallest eigenvalues, we check which subproblems the remaining eigenvalues belong to. If there are no eigenvalues left for some subproblem, the number of computed eigenvalues for this subproblem is probably not enough. Subsequently we restart computing this subproblem with an increasing number of required eigenvalues.

5.3 Two-level parallel implementation

We have addressed in Section 3 that the n_c decomposed problems are independent to each other and can be solved simultaneously. Accordingly we have a two-level parallel implementation illustrated by Figure 1. At the first level, subproblems are dispatched among groups of processors. At the second level, the irreducible grids are distributed among each group of processors. Since eigenfunctions of different subproblems are naturally orthogonal, there is no communication between different groups of processors during solving the eigenvalue problem. Such two-level or multi-level parallelism is likely appreciable for the architecture hierarchy of modern supercomputer. In Section 6.3, it will be shown that the two-level parallel implementation reduces the communication cost.

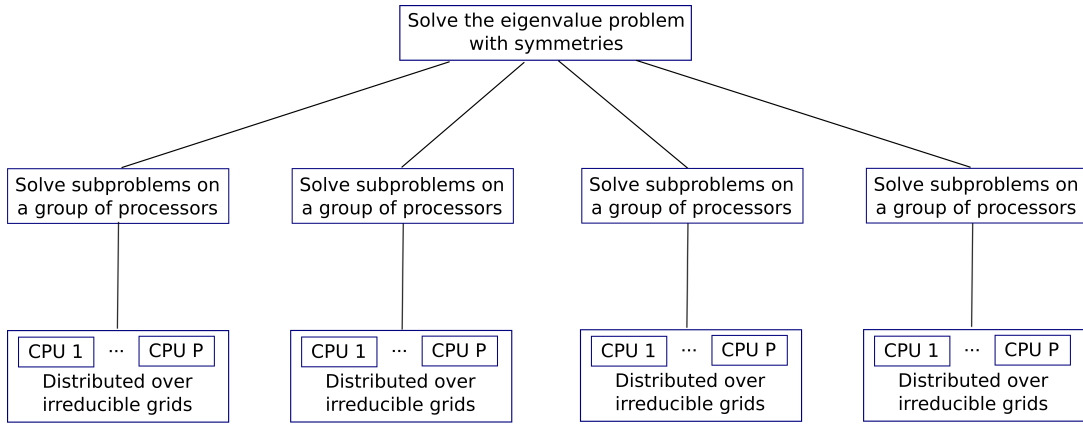


Figure 1: Schematic illustration for two-level parallel implementation for solving the eigenvalue problem with point symmetries. *Actually, the number of processors of each group can be in proportion to $d_\nu N_0$, the size of discretized basis for each subproblem.*

6 Numerical tests and applications

In this section, we present some numerical examples arising from eigenvalue problems in quantum mechanics to validate the implementation and illustrate the efficiency of the decomposition approach. We use hexahedral finite elements for discretizations and consider crystallographic point groups of which symmetry operations keep the hexahedral grids invariant. The matrix eigenvalue problem is solved by subroutines of ARPACK. Our computing platform is the LSSC-III cluster provided by State Key Laboratory of Scientific and Engineering Computing (LSEC).

6.1 Validation of implementation

First we validate the implementation of the decomposition approach. Consider the harmonic oscillator equation which is a basic quantum eigenvalue problem as like:

$$-\frac{1}{2}\Delta u + \frac{1}{2}|x|^2 u = \lambda u \quad \text{in } \mathbb{R}^3. \quad (6.1)$$

The exact eigenvalues are given as

$$\lambda_{k,m,n} = k + m + n + 1.5, \quad k, m, n = 0, 1, 2, \dots$$

In this example, we solve the first 10 eigenvalues. The computation can be done in a finite domain with zero boundary condition since the eigenfunctions decay exponentially.

It is obvious that the system is spherically symmetric. As representatives, we test Abelian subgroup D_{2h} and non-Abelian subgroups D_4 and D_{2d} . Table 2 gives the irreducible representation matrices of these groups. We refer to Appendix A for a detailed description about the notation in the table. The hexahedral grids can be kept invariant under the three groups.

Table 2: Representation matrices of point symmetry groups D_{2d} , D_{2h} and D_4

D_{2h}	E	C_{2x}	C_{2y}	C_{2z}	I	IC_{2x}	IC_{2y}	IC_{2z}
$\Gamma^{(1)}$	1	1	1	1	1	1	1	1
$\Gamma^{(2)}$	1	1	-1	-1	1	1	-1	-1
$\Gamma^{(3)}$	1	-1	1	-1	1	-1	1	-1
$\Gamma^{(4)}$	1	-1	-1	1	1	-1	-1	1
$\Gamma^{(5)}$	1	1	1	1	-1	-1	-1	-1
$\Gamma^{(6)}$	1	1	-1	-1	-1	-1	1	1
$\Gamma^{(7)}$	1	-1	1	-1	-1	1	-1	1
$\Gamma^{(8)}$	1	-1	-1	1	-1	1	1	-1
D_{2d}	E	C_{2y}	IC_{4y}	IC_{4y}^{-1}	IC_{2x}	IC_{2z}	C_{2c}	C_{2d}
$\Gamma^{(1)}$	1	1	1	1	1	1	1	1
$\Gamma^{(2)}$	1	1	-1	-1	1	1	-1	-1
$\Gamma^{(3)}$	1	1	1	1	-1	-1	-1	-1
$\Gamma^{(4)}$	1	1	-1	-1	-1	-1	1	1
$\Gamma^{(5)}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$
D_4	E	C_{2y}	C_{4y}	C_{4y}^{-1}	C_{2x}	C_{2z}	C_{2c}	C_{2d}
$\Gamma^{(1)}$	1	1	1	1	1	1	1	1
$\Gamma^{(2)}$	1	1	-1	-1	1	1	-1	-1
$\Gamma^{(3)}$	1	1	1	1	-1	-1	-1	-1
$\Gamma^{(4)}$	1	1	-1	-1	-1	-1	1	1
$\Gamma^{(5)}$	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$

According to Theorem 2.1, we can decompose the original eigenvalue problem (6.1) as follows:

1. Applying D_{2d} , we have 8 completely decoupled subproblems;

2. Applying D_4 or D_{2d} , we have 6 subproblems and two of them corresponding to representation $\Gamma^{(5)}$ are coupled eigenvalue problems.

We employ trilinear finite elements to solve these eigenvalue subproblems, and tell from the convergence rate of eigenvalues that the implementation is correct. Taking non-Abelian group D_4 for instance, we exhibit in Figure 2 errors in eigenvalue approximations obtained from solving the subproblems. And the h^2 -convergence rate can be observed.

Moreover, in Table 3, we list the ν - l symmetries of computed eigenfuntions from solving the subproblems. It can be seen that degenerate eigenvalues are distributed over subproblems.

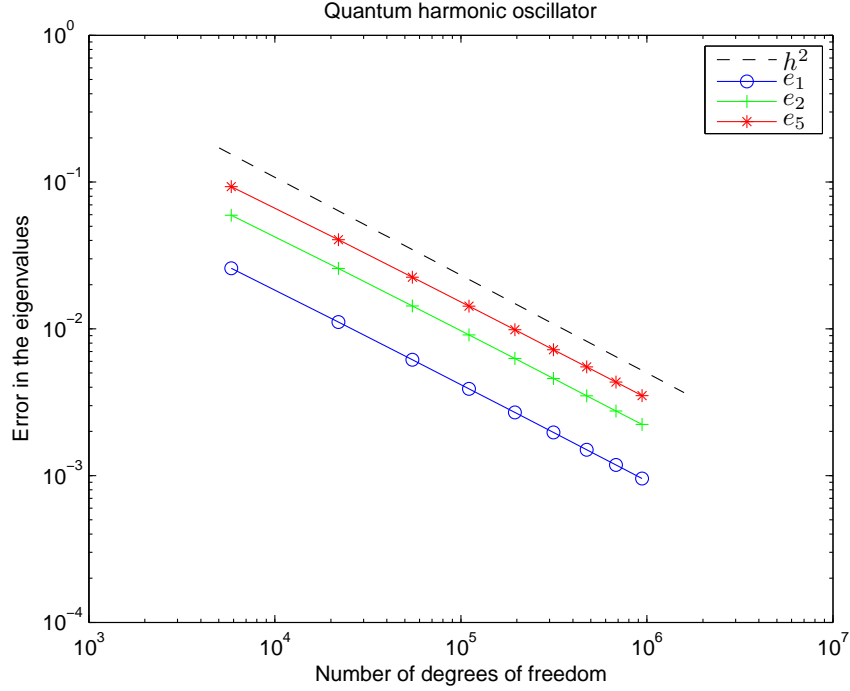


Figure 2: Error in the eigenvalue approximations from solving 6 subproblems associated with non-Abelian group D_4 . Errors in the first three different eigenvalues are labeled as e_1 , e_2 , e_5 respectively. The h^2 -convergence rate can be observed.

6.2 Reduction in computational cost

Taking Abelian group D_{2h} as an example, we compare the computational cost of solving the original eigenvalue problem (6.1) with that of solving 8 subproblems. The first 110 eigenvalues of the original eigenvalue problem are needed. And it is sufficient to solve the first 22 eigenvalues of each subproblem. In order to illustrate and analyze the saving in computational cost, we launch the tests on a single CPU core.

In Table 4, we present statistics from trilinear finite element discretizations. The average CPU time of a single iteration during solving the original problem (6.1) is 42.29 seconds while that of solving 8 subproblems is 3.61 seconds². In Table 5, we present statistics from tricubic finite elements. The average CPU time of a single iteration during solving the original problem (6.1) is 60.80 seconds while that of solving 8 subproblems is 9.71 seconds.

² We count the average CPU time of a single iteration for each subproblem and then accumulate them. Taking

Table 3: The ν - l symmetries of the first 10 eigenfunctions of problem (6.1)

	D _{2h}		D ₄		D _{2d}	
	ν	l	ν	l	ν	l
u_1	1	1	1	1	1	1
u_2	8	1	3	1	5	1
u_3	6	1	5	1	5	2
u_4	7	1	5	2	2	1
u_5	3	1	5	1	4	1
u_6	4	1	5	2	5	1
u_7	2	1	4	1	5	2
u_8	1	1	1	1	1	1
u_9	1	1	2	1	2	1
u_{10}	1	1	1	1	1	1

Table 4: Statistics of solving the original problem (6.1) and 8 subproblems using trilinear finite elements. *In Column 1, subproblems are labeled by different ν s. In Columns 2 and 3, we list the number of major iteration steps and matrix-vector multiplications. In Columns 4 and 5, we present CPU time spent on matrix-vector multiplications and the whole procedure of IRLM.*

Problem	#iter.	#OP*x	time_mv (sec.)	time_total (sec.)
(6.1)	22	1599	175.01	930.39
$\nu = 1$	18	356	5.13	8.21
$\nu = 2$	22	420	6.16	9.95
$\nu = 3$	22	421	6.06	9.83
$\nu = 4$	21	406	5.84	9.46
$\nu = 5$	18	353	5.06	8.23
$\nu = 6$	21	405	5.74	9.44
$\nu = 7$	20	389	5.51	9.03
$\nu = 8$	21	403	5.75	9.34

Table 5: Statistics of solving (6.1) and 8 subproblems using tricubic finite elements.

Problem	#iter.	#OP*x	time_mv (sec.)	time_total (sec.)
(6.1)	57	3972	1696.29	3465.57
$\nu = 1$	50	937	55.15	62.75
$\nu = 2$	64	1156	67.75	77.11
$\nu = 3$	64	1153	67.57	77.09
$\nu = 4$	62	1128	66.05	75.25
$\nu = 5$	47	892	52.18	59.31
$\nu = 6$	69	1215	71.15	81.37
$\nu = 7$	63	1134	66.74	76.12
$\nu = 8$	70	1230	72.46	82.87

The speedup in average CPU time of a single iteration is 11.72 with trilinear finite elements while it is decreased to 6.26 with tricubic finite elements. This numerical phenomenon can be explained by performance analysis (4.6). In our computation, the maximum dimension of Krylov subspace is twice the number of required eigenvalues plus 5, which is recommended by ARPACK’s tutorial examples. So we have $\theta_1 = 4.59$. It can be calculated from the statistics of solving the original problem that the CPU time percentage ω of matrix-vector multiplications is 0.19 with trilinear finite elements and grows to 0.59 with tricubic finite elements. Correspondingly, using (4.6), we predict that the CPU time speedup would be 12.58 and 7.64 respectively.

According to (4.3), the computational cost of QR -iteration grows faster than that of matrix-vector multiplication when the number of required eigenvalues increases. Thus we can see that the decomposition approach would be more appreciable for large-scale eigenvalue problems.

6.3 Saving in communication

Besides the reduction in computational cost, solving decoupled problems will also save communication among parallel processors. As mentioned in Section 5.3, our implementation of the decomposition approach is parallelized in two levels. The DOFs are distributed in each group of processors and no communication occurs between any two groups of processors. This leads to a saving in communication.

We still consider the oscillator eigenvalue problem (6.1) and decompose it into 8 subproblems according to symmetry group D_{2h} . The comparison of communication between solving the original problem and the subproblems is given in Table 6.

Table 6: Comparison of communication between solving (6.1) and 8 subproblems. *Column 1 gives the number of processors N_p . In the other columns, “use symm” corresponds to solving the subproblems and “not use” corresponds to solving the original eigenvalue problem. Columns 2 and 3 give the average number of processors each processor communicates with. Columns 4 and 5 list the average number of Bytes sent by each processor. And the last two columns report the CPU time spent on communication during matrix-vector multiplications.*

N_p	N_p in comm		Bytes in comm		CPU time in comm (sec.)	
	use symm	not use	use symm	not use	use symm	not use
8	0.00	1.75	0	134,560	0.00	8.93
16	1.00	1.88	19,608	145,451	0.20	10.36

6.4 Applications to Kohn–Sham equations

Now we apply the decomposition approach to electronic structure calculations of symmetric molecules. In the context of density functional theory (DFT), ground state properties of many-electron systems are usually obtained by solving the Kohn–Sham equation [27, 31, 35]. It is a nonlinear eigenvalue problem as follows

$$\left(-\frac{1}{2}\Delta + V^{\text{eff}}[\rho]\right) \Psi_n = \epsilon_n \Psi_n \quad \text{in } \mathbb{R}^3, \quad (6.2)$$

Table 4 for example, we have that $3.61 = \frac{8.21}{18} + \frac{9.95}{22} + \frac{9.83}{22} + \frac{9.46}{21} + \frac{8.23}{18} + \frac{9.44}{21} + \frac{9.03}{20} + \frac{9.34}{21}$

where $\rho(\mathbf{r}) = \sum_{n=1}^{N_e} f_n |\Psi_n(\mathbf{r})|^2$ is the charge density contributed by N_e eigenfunctions $\{\Psi_n\}$ with occupancy numbers $\{f_n\}$, and $V^{\text{eff}}[\rho]$ the so-called effective potential which is a nonlinear functional of ρ . On the assumption of no external fields, $V^{\text{eff}}[\rho]$ can be written into

$$V^{\text{eff}} = V^{\text{ne}} + V^{\text{H}} + V^{\text{xc}}, \quad (6.3)$$

where V^{ne} is the Coulomb potential between the nuclei and the electrons, V^{H} the Hartree potential, and V^{xc} the exchange-correlation potential [35]. The ground state density of a confined system decays exponentially [2, 23, 41], so we choose the computational domain as an appropriate cube and impose zero boundary condition.

As a nonlinear eigenvalue problem, Kohn–Sham equation (6.2) is solved by the self-consistent field (SCF) iteration [35]. The dominant part of computation is the repeated solving of the linearized Kohn–Sham equation with a fixed effective potential. The number of required eigenstates grows in proportion to the number of valence electrons in the system. Therefore the Kohn–Sham equation solver will probably make the performance bottleneck for large-scale DFT calculations.

Real-space discretization methods are attractive for confined systems since they allow a natural imposition of the zero boundary condition [6, 32]. Among real-space mesh techniques, the finite element method keeps both locality and the variational property, and has been applied to electronic structure calculations (see, e.g., [1, 18, 19, 39, 40, 42, 43, 47, 48, 51]); others like the finite difference method, finite volume method and the wavelet approach have also shown the potential in this field [13, 16, 22, 26, 29, 32, 38].

We solve the Kohn–Sham equation of some symmetric molecules with tricubic finite element discretizations. The statistics are summarized in Table 7. The full symmetry group of these molecules is T_d . For simplicity we select subgroup D_2 . Accordingly, the Kohn–Sham equation can be decomposed into 4 decoupled subproblems. It is indicated by Table 7 that the decomposition approach is appreciable for large-scale symmetry molecular systems.

Table 7: Comparison between solving the original Kohn–Sham equation and subproblems. *Column 3 gives the number of required eigenstates. The number of DOFs given in Columns 4 and 5 is required by the convergence of ground state energy [18]. Columns 7 and 8 list the average CPU time in diagonalization at each SCF iteration step, which is the dominant part of time. The last column is the speedup of the decomposition approach.*

System	G	N_e	DOFs		N_p	CPU time in diag. (sec.)		Speedup
			not use	use symm		not use	use symm	
C ₁₂₃ H ₁₀₀	D ₂	300	1,191,016	297,754	32	2,783	558	4.99
C ₂₇₅ H ₁₇₂	D ₂	640	1,643,032	410,758	32	13,851	1,559	8.88
C ₅₂₅ H ₂₇₆	D ₂	1200	2,097,152	524,288	64	25,296	2,334	10.84

7 Concluding remarks

In this paper, we propose a decomposition approach to eigenvalue problems with spatial symmetries. Different from the classical treating of symmetries in quantum chemistry, our approach does not explicitly construct the symmetry-adapted linear combinations (SALCs). Instead, we formulate a set of eigenvalue subproblems easy for grid-based discretizations. Moreover, we

provide mathematical guidance to the construction of SALCs and obtain the exact relation between the two approaches.

Such a decomposition can reduce the computational cost remarkably since a smaller number of eigenpairs are solved for each subproblem. The quantization of this reduction implies that our approach could be appreciable for large-scale eigenvalue problems. In practice, we solve a sufficient number of redundant eigenpairs for each subproblem in order not to miss any eigenpairs. It would be very helpful for reducing the extra work if one could predict the distribution of eigenpairs among subproblems.

Under finite element discretizations, this decomposition method has been applied to Kohn-Sham equations of symmetric molecules. If solving Kohn-Sham equations of periodic crystals, we should consider plane wave expansion which could be regarded as grid-based discretization in reciprocal space. In Appendix C, we will show that the invariance under some coordinate transformation can be kept by Fourier transformation. So the decomposition approach is also applicable to plane waves.

In numerical examples, we treat only a part of cubic symmetries for validation and illustration. Obviously, the decomposition approach and its practical issues can be adapted to other spatial symmetries with appropriate grids.

In this paper, we concentrate on spatial symmetries. It is possible to use other symmetries in some specific areas. For instance, [20, 36] exploit the angular momentum, spin and parity symmetries of atoms during solving the Schrödinger equation. It is our ongoing work to exploit these underlying or internal symmetries.

ACKNOWLEDGEMENTS. The authors would like to thank Prof. Xiaoying Dai, Prof. Xingao Gong, Prof. Lihua Shen, Dr. Zhang Yang and Mr. Jinwei Zhu for their stimulating discussions on electronic structure computations. The second author is grateful to Prof. Zeyao Mo for his encouragement.

Appendix A: Basic concept of group theory

In this appendix, we include some basic concepts of group theory for a more self-contained exposition. They could be found in standard textbooks like [30, 44, 49].

A group G is a set of elements $\{R\}$ with a well-defined multiplication operation which satisfy several requirements:

1. The set is closed under the multiplication.
2. The associative law holds.
3. There exists a unit element E such that $ER = RE = R$ for any $R \in G$.
4. There is an inverse R^{-1} in G to each element R such that $RR^{-1} = R^{-1}R = E$.

Group G is called finite if it contains a finite number of elements. And this number, denoted by g , is said to be the order of the group. The rearrangement theorem tells that the elements of G are only rearranged by multiplying each by any $R \in G$, i.e., $RG = G$ for any $R \in G$.

An element $R_1 \in G$ is called to be conjugate to R_2 if $R_2 = SR_1S^{-1}$, where S is some element in the group. All the mutually conjugate elements form a class of elements. It can be proved that group G can be divided into distinct classes. Denote the number of classes as n_c .

Two groups is called to be homomorphic if there exists a correspondence between the elements of the two groups as $R \leftrightarrow R'_1, R'_2, \dots$, which means that if $RS = T$ then the product

of any R'_i with any S'_j will be a member of the set $\{T'_1, T'_2, \dots\}$. In general, a homomorphism is a many-to-one correspondence. It specializes to an isomorphism if the correspondence is one-to-one.

A representation of a group is any group of mathematical entities which is homomorphic to the original group. We restrict the discussion to matrix representations. Any matrices representation with nonvanishing determinants is equivalent to a representation by unitary matrices. Two representations are said to be equivalent if they are associated by a similarity transformation. If a representation can not be equivalent to representations of lower dimensionality, it is called irreducible.

The number of all the inequivalent, irreducible, unitary representations is equal to the number of classes in G . The Celebrated Theorem tells that

$$\sum_{\nu=1}^{n_c} d_\nu^2 = g, \quad (\text{A.1})$$

where d_ν denotes the dimensionality of the ν -th representation.

Finally we introduce some crystallographic point groups used in our examples. Groups D_{2h} , D_{2d} and D_4 are three dihedral groups; the first one is Abelian and the other two are non-Abelian. In Table 2, C_{nj} denotes a rotation about axis Oj by $2\pi/n$ in the right-hand screw sense and I is the inversion operator [14].

Appendix B: Proof of Proposition 2.2

Proof. (a) Since $\{P_R\}$ are unitary operators, we have

$$\mathcal{P}_{ml}^{(\nu)*} = \left(\frac{d_\nu}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* P_R \right)^* = \frac{d_\nu}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml} P_{R^{-1}} = \frac{d_\nu}{g} \sum_{S \in G} \Gamma^{(\nu)}(S^{-1})_{ml} P_S,$$

which together with the fact that $\Gamma^{(\nu)}$ is a unitary representation derive

$$\mathcal{P}_{ml}^{(\nu)*} = \frac{d_\nu}{g} \sum_{S \in G} \Gamma^{(\nu)}(S)_{lm}^* P_S = \mathcal{P}_{lm}^{(\nu)}.$$

(b) According to the definition

$$\begin{aligned} \mathcal{P}_{ml}^{(\nu)} \mathcal{P}_{m'l'}^{(\nu')} &= \left(\frac{d_\nu}{g} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* P_R \right) \left(\frac{d_{\nu'}}{g} \sum_{S \in G} \Gamma^{(\nu')}(S)_{m'l'}^* P_S \right) \\ &= \frac{d_\nu d_{\nu'}}{g^2} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \left(\sum_{S \in G} \Gamma^{(\nu')}(S)_{m'l'}^* P_{RS} \right). \end{aligned}$$

Note that the rearrangement theorem implies that, when S runs over all the group elements, $S' = RS$ for any R also runs over all the elements. Hence we have

$$\begin{aligned} \mathcal{P}_{ml}^{(\nu)} \mathcal{P}_{m'l'}^{(\nu')} &= \frac{d_\nu d_{\nu'}}{g^2} \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \left(\sum_{S' \in G} \Gamma^{(\nu')}(R^{-1}S')_{m'l'}^* P_{S'} \right) \\ &= \frac{d_\nu d_{\nu'}}{g^2} \sum_{S' \in G} \left(\sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \Gamma^{(\nu')}(R^{-1}S')_{m'l'}^* \right) P_{S'}. \end{aligned}$$

We may calculate as follows

$$\begin{aligned}
\sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \Gamma^{(\nu')}(R^{-1}S')_{m'l'}^* &= \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \left(\sum_{n=1}^{d_{\nu'}} \Gamma^{(\nu')}(R^{-1})_{m'n}^* \Gamma^{(\nu')}(S')_{nl'}^* \right) \\
&= \sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \left(\sum_{n=1}^{d_{\nu'}} \Gamma^{(\nu')}(R)_{nm'} \Gamma^{(\nu')}(S')_{nl'}^* \right) \\
&= \sum_{n=1}^{d_{\nu'}} \Gamma^{(\nu')}(S')_{nl'}^* \left(\sum_{R \in G} \Gamma^{(\nu)}(R)_{ml}^* \Gamma^{(\nu')}(R)_{nm'} \right) \\
&= \delta_{\nu\nu'} \delta_{lm'} \frac{g}{d_{\nu'}} \Gamma^{(\nu)}(S')_{ml'}^*,
\end{aligned}$$

where the last equality is derived from the great orthogonality theorem. Thus we arrive at

$$\mathcal{P}_{ml}^{(\nu)} \mathcal{P}_{m'l'}^{(\nu')} = \delta_{\nu\nu'} \delta_{lm'} \frac{d_{\nu}}{g} \sum_{S' \in G} \Gamma^{(\nu)}(S')_{ml'}^* P_{S'} = \delta_{\nu\nu'} \delta_{lm'} \mathcal{P}_{ml'}^{(\nu)}.$$

□

Appendix C: Spatial symmetry in reciprocal space

Plane wave method is widely used for solving the Kohn–Sham equations of crystals. Actually, plane waves are regarded as grid-based discretizations in reciprocal space. We will show that the symmetry equation in real space is kept in reciprocal space. The solution domain Ω of crystals can be spanned by three lattice vectors in real space. On the assumption that a function f is invariant with integer multiple translations of the lattice vectors, we present the function in reciprocal space as like:

$$f(\mathbf{q}) = \frac{1}{N} \sum_{\mathbf{r}} f(\mathbf{r}) e^{-i\mathbf{q} \cdot \mathbf{r}}, \quad (\text{C.1})$$

where N is the number of grid points. If f is kept invariant with the coordinate transform R in Ω , then

$$\begin{aligned}
f(R\mathbf{q}) &= \frac{1}{N} \sum_{\mathbf{r}} f(\mathbf{r}) e^{-i(R\mathbf{q}) \cdot \mathbf{r}} \\
&= \frac{1}{N} \sum_{\mathbf{r}} f(\mathbf{r}) e^{-i\mathbf{q} \cdot (R^{-1}\mathbf{r})} \\
&= \frac{1}{N} \sum_{R^{-1}\mathbf{r}} f(R^{-1}\mathbf{r}) e^{-i\mathbf{q} \cdot (R^{-1}\mathbf{r})} \\
&= f(\mathbf{q}).
\end{aligned}$$

The above second equation holds since the coordinate transformation R can be represented as an orthogonal matrix. And the above third one holds due to the assumption that

$$f(R\mathbf{r}) = f(\mathbf{r}), \quad \forall \mathbf{r} \in \Omega.$$

Thus the decomposition method is probably applicable to plane waves.

References

- [1] J. Ackermann, B. Erdmann, and R. Roitzsch. A self-adaptive multilevel finite element method for the stationary Schrödinger equation in three space dimensions. *J. Chem. Phys.*, 101:7643–7650, 1994.
- [2] S. Agmon. *Lectures on the Exponential Decay of Solutions of Second-Order Elliptic Operators*. Princeton University Press, Princeton, NJ, 1981.
- [3] I. Babuska and J. E. Osborn. Finite element-Galerkin approximation of the eigenvalues and eigenvectors of self-adjoint problems. *Math. Comput.*, 52(186):275–297, 1989.
- [4] I. Babuska and J. Osborn. Eigenvalue problems. In *Handbook of Numerical Analysis*, volume II, pages 641–787. North-Holland, 1991.
- [5] L. Banjai. Eigenfrequencies of fractal drums. *J. Comput. Appl. Math.*, 198:1–18, 2007.
- [6] T. L. Beck. Real-space mesh techniques in density-functional theory. *Rev. Mod. Phys.*, 72:1041–1080, 2000.
- [7] J. K. Bennighof and R. B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM J. Sci. Comput.*, 25:2084–2106, 2004.
- [8] D. M. Bishop. *Group Theory and Chemistry*. Dover, New York, 1993.
- [9] A. Bossavit. Symmetry, groups, and boundary value problems. A progressive introduction to noncommutative harmonic analysis of partial differential equations in domains with geometrical symmetry. *Comp. Meth. Appl. Mech. Engng.*, 56:167–215, 1986.
- [10] A. Bossavit. Boundary value problems with symmetry, and their approximation by finite elements. *SIAM J. Appl. Math.*, 53:1352–80, 1993.
- [11] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. Le Bris, and Y. Mada. Computational quantum chemistry: a primer. In Ph. G. Ciarlet and C. Le Bris, editors, *Handbook of Numerical Analysis, Special volume, Computational Chemistry, Volume X*, pages 3–270. North-Holland, 2003.
- [12] F. Chatelin, *Spectral Approximations of Linear Operators*, Academic Press, New York, 1983.
- [13] J. R. Chelikowsky, N. Troullier, and Y. Saad. Finite-difference-pseudopotential method: Electronic structure calculations without a basis. *Phys. Rev. Lett.*, 72:1240–1243, 1994.
- [14] J. F. Cornwell. *Group Theory in Physics: An Introduction*. Academic Press, California, 1997.
- [15] F. A. Cotton. *Chemical Applications of Group Theory*. John Wiley and Sons, New York, 3rd edition, 1990.
- [16] X. Dai, X. Gong, Z. Yang, D. Zhang, and A. Zhou. Finite volume discretizations for eigenvalue problems with applications to electronic structure calculations. *Multiscale Model. Simul.*, 9:208–240, 2011.

- [17] X. Dai, Z. Yang, and A. Zhou, *Symmetric finite volume schemes for eigenvalue problems in arbitrary dimensions*, Sci. China Ser. A, 51:1401–1414, 2008.
- [18] J. Fang, X. Gao, and A. Zhou. A Kohn–Sham equation solver based on hexahedral finite elements. *J. Comput. Phys.*, 231:3166–3180, 2012.
- [19] J.-L. Fattebert, R. D. Hornung, and A. M. Wissink. Finite element approach for density functional theory calculations on locally-refined meshes. *J. Comput. Phys.*, 223:759–773, 2007.
- [20] G. Friesecke and B. D. Goddard. Asymptotics-based CI models for atoms: properties, exact solution of a minimal model for Li to Ne, and application to atomic spectra. *Multiscale Model. Simul.*, 7:1876–1897, 2009.
- [21] W. Gao, X. S. Li, C. Yang, and Z. J. Bai. An implementation and evaluation of the AMLS method for sparse eigenvalue problems. *ACM Trans. Math. Soft.*, 5:1–27, 2007.
- [22] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider. Daubechies wavelets as a basis set for density functional pseudopotential calculations. *J. Chem. Phys.*, 129:014109, 2008.
- [23] D. Gilbarg and N. S. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Springer, Berlin, Heidelberg, 3rd edition, 2001.
- [24] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.
- [25] W. Hackbusch. *Elliptic Differential Equations: Theory and Numerical Treatment*. Springer-Verlag, Berlin Heidelberg, 1992.
- [26] Y. Hasegawa, J.-I. Iwata, M. Tsuji, D. Takahashi, A. Oshiyama, K. Minami, T. Boku, F. Shoji, A. Uno, M. Kurokawa, H. Inoue, I. Miyoshi, and M. Yokokawa. First-principles calculations of electron states of a silicon nanowire with 100,000 atoms on the K computer. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC2011)*, pages 1–11, 2011.
- [27] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev. B*, 136(3B):B864–B871, 1964.
- [28] W. C. Hurty. Vibrations of structure systems by component-mode synthesis. *ASCE J. Engng. Mech. Division*, 86:51–69, 1960.
- [29] J.-I. Iwata, D. Takahashi, A. Oshiyama, T. Boku, K. Shiraishi, S. Okada, and K. Yabana. A massively-parallel electronic-structure calculations based on real-space density functional theory. *J. Comput. Phys.*, 229:2339–2363, 2010.
- [30] H. Jones. *The Theory of Brillouin Zones and Electronic States in Crystals*. North-Holland, Amsterdam, 1960.
- [31] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4A):A1133–A1138, 1965.

- [32] L. Kronik, A. Makmal, M. L. Tiago, M. M. G. Alemany, M. Jain, X. Huang, Y. Saad, and J. R. Chelikowsky. Parsec – the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structures. *Phys. Stat. Sol. (b)*, 243:1063–1079, 2006.
- [33] J. R. Kuttler and V. G. Sigillito. Eigenvalues of the Laplacian in two dimensions. *SIAM Rev.*, 26:163–193, 1984.
- [34] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users’ Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM, Philadelphia, 1998.
- [35] R. M. Martin. *Electronic Structure: Basic Theory and Practical Methods*. Cambridge University Press, Cambridge, 2004.
- [36] C. B. Mendl and G. Friessecke. Efficient algorithm for asymptotics-based configuration-interaction methods and electronic structure of transition metal atoms. *J. Chem. Phys.*, 133:184101, 2010.
- [37] J. M. Neuberger, N. Sieben, and J. W. Swift. Computing eigenfunctions on the Koch Snowflake: A new grid and symmetry. *J. Comput. Appl. Math.*, 191:126–142, 2006.
- [38] T. Ono and K. Hirose. Real-space electronic-structure calculations with a time-saving double-grid technique. *Phys. Rev. B*, 72:085115, 2005.
- [39] J. E. Pask, B. M. Klein, C. Y. Fong, and P. A. Sterne. Real-space local polynomial basis for solid-state electronic-structure calculations: A finite-element approach. *Phys. Rev. B*, 59:12352–11358, 1999.
- [40] J. E. Pask and P. A. Sterne. Finite element methods in *ab initio* electronic structure calculations. *Model. Simul. Mater. Sci. Eng.*, 13:71–96, 2005.
- [41] B. Simon. Schrödinger operators in the twentieth century. *J. Math. Phys.*, 41:3523–3555, 2000.
- [42] P. A. Sterne, J. E. Pask, and B. M. Klein. Calculation of positron observables using a finite element-based approach. *Appl. Surf. Sci.*, 149:238–243, 1999.
- [43] P. Suryanarayana, V. Gavini, and T. Blesgen. Non-periodic finite-element formulation of Kohn-Sham density functional theory. *J. Mech. Phys. Solids*, 58:256–280, 2010.
- [44] M. Tinkham. *Group Theory and Quantum Mechanics*. McGraw-Hill, New York, 1964.
- [45] T. Torsti, T. Eirola, J. Enkovaara, T. Hakala, P. Havu, V. Havu, T. Höynälänmaa, J. Ignatius, M. Lyly, I. Makkonen, T. T. Rantala, J. Ruokolainen, K. Ruotsalainen, E. Räsänen, H. Saarikoski, and M. J. Puska, *Three real-space discretization techniques in electronic structure calculations*, *Phys. Stat. Sol.*, B243:1016-1053, 2006.
- [46] L. N. Trefethen and T. Betcke. Computed eigenmodes of planar regions. In *Recent advances in differential equations and mathematical physics*, volume 412 of *Contemp. Math.*, pages 297–314, Providence, RI, 2006. Amer. Math. Soc.
- [47] E. Tsuchida and M. Tsukada. Electronic-structure calculations based on the finite-element method. *Phys. Rev. B*, 52:5573–5578, 1995.

- [48] S. R. White, J. W. Wilkins, and M. P. Teter. Finite-element method for electronic structure. *Phys. Rev. B*, 39:5819–5833, 1989.
- [49] E. P. Wigner. *Group Theory and its Application to the Quantum Mechanics of Atomic Spectra*. Academic Press, New York, 1959.
- [50] D. C. Young. *Computational Chemistry: A Practical Guide for Applying Techniques to Real-World Problems*. John Wiley and Sons, New York, 2001.
- [51] D. Zhang, L. Shen, A. Zhou, and X. Gong. Finite element method for solving Kohn-Sham equations based on self-adaptive tetrahedral mesh. *Phys. Lett. A*, 372:5071–5076, 2008.